Visual Stimulation Software v2.4 Documentation

New features in v2.4 are highlighted. See changelog for more details of update history.

1. Introduction

The current version of the software is based on Bruno's scripts. It is designed as a single Matlab function, '*VisStimAlex*' with numerous parameters that can be changed as needed, such as spatial or temporal frequency. All parameters can be inputted as "real world values" – so, for example, spatial frequency is inputted as the angular frequency subtended on the mouse's retina. These parameters are processed in to a format suitable for the computer, and then *VisStimAlex* calls one of ten separate 'stimulus generating functions'. This way of doing things means there's quite a lot of code, which can be a bit of a headache to edit, but since the code is all predefined it executes pretty quickly and predictably.

These considerations are not important for using the software as they are all concealed 'under the hood'. The most important parameters within *VisStimAlex* are *experiment type* and *triggering*. See below for more details.

When the function is run, the screen will turn white. When it is ready to start, the screen will turn grey, and execution pauses for a keypress or trigger. The screen will then turn black, and the stimuli will be shown. Once the experiment has finished, the screen will turn black again.

VisStimAlex automatically saves information about the stimulus in a single struct variable called '*stimulusInfo*', the contents of which depend on the experiment that was run. This struct contains information about the experimental parameters, including measured timings for each state. It is saved automatically with a filename consisting of the date and time of the start of the experiment.

2. Using VisStimAlex

1. To run with default parameters, use: *[stimulusInfo, filePath] = VisStimAlex*
    i. The stimulusInfo struct will be saved to disk automatically, as described above. VisStimAlex then outputs the complete file path, as well as stimulusInfo itself.
2. To change a parameter's default value, simply edit it in *VisStimAlex*
3. Parameter values can also be specified for a single experiment, without needing to change the default value. This is done by calling *VisStimAlex* with parameter values specified as optional arguments:

    [stimulusInfo, filePath] = VisStimAlex('Parameter', 'Value', …)

3. Displaying information about the stimulation

For a quick impression of how the stimulation has performed, call the function

parseStimulus(stimulusInfo)

This function analyses the stimulus information and displays a 'stimulus timeline' – showing which stimulus was displayed when. It also displays information such as the variability of the timings, which gives an indicator of whether the stimulus program was working properly. Under ideal conditions this variation should be less than the duration of a display frame (16.6ms for a 60 Hz monitor).

4. Experiment types

Currently, there are five experiment types supported. These are:

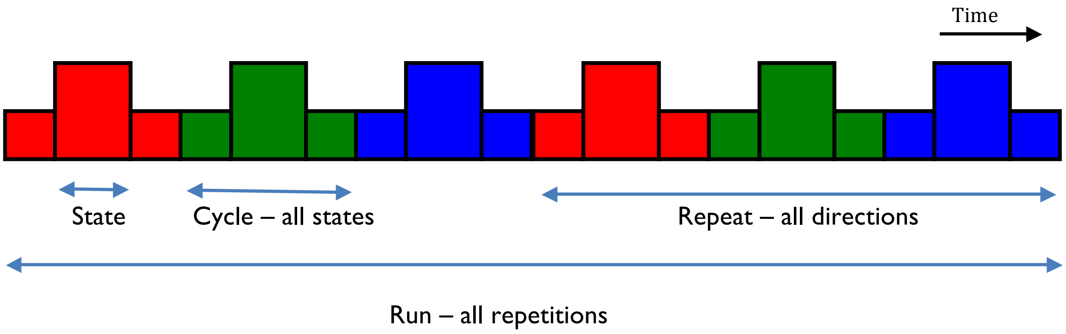
1. 'Flip' – flashing the screen between black and white. Useful for maximally stimulating an eye, or testing equipment.
2. 'D' – drifting gratings

3. 'DH' – a drifting grating which is then held stationary; then the next direction is shown (drifting first)
4. 'HD' – a stationary grating which then drifts; then the next direction is shown (stationary first)
5. 'HDH' – a stationary grating which then drifts, and is then held again; then the next direction is shown

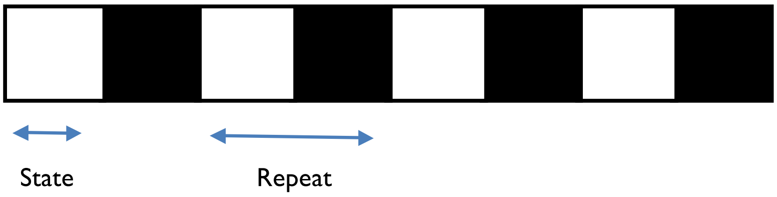
The exact behaviour of these modes depends on whether triggering is on or not. See section 6

5. States, Cycles, Repetitions and Runs

A '*run'* is a complete experiment. Each run consists of a number of '*repeats'*, each of which has a number of '*directions'* ('*cycles'*), each of which has a number of '*states'* (drifting or stationary gratings, or black or white for *Flip* experiments):



A hold state is represented by a lower height rectangle; a drift by a higher rectangle. Direction is represented by the colour of the rectangle (i.e. red could be 0°, green 120° and blue 240°). So this experimental run has 2 repeats of 3 different direction. Each direction cycle has three states – a hold, a drift, and a hold.



In this case, a flip program, the *run* consists of 4 *repeats*. Since there are no directions in the flip program (as there are no gratings), each repeat simply contains two states – white and black. This is a special case – each direction cycle consists of one state only.

6. Triggering

Triggering can be set to 'off', in which case stimuli are presented for a period of time determined by parameters within the VisStimAlex script. This is not recommended. Triggering set 'on' causes the stimulation to continue until a trigger (TTL High) is detected.

A trigger is only needed at the end of each *direction cycle*. This means a trigger is not needed after each state. When triggering is on, the timing parameter for the last state is not used; rather the script simply continues to display this state until a trigger is detected. So, for example, for mode 'HD' with triggering on, a stationary grating is displayed for a defined period of time (defined by the parameter 'preDriftHoldTime'), after which it drifts. When a trigger is detected, the next grating is displayed.

This trigger should be at least twice the duration of a stimulus frame to ensure VisStimAlex, which only polls for trigger state after each frame, does not miss it. So for a 60 Hz monitor the trigger should be at least 33ms; obviously using something like a 50ms trigger is safer. Using long triggers should not have any negative effects (unless they start to approach the duration of a direction cycle).

7. Changing program flow during execution

It is possible to quit VisStimAlex whilst it is being run, by pressing and holding the escape key. Program execution will end as soon as possible. The function will still return stimulusInfo, and will save it automatically just as if it were allowed to run to completion.

It is also possible to simulate a trigger by pressing the ‘t’ key (make sure caps lock is off).

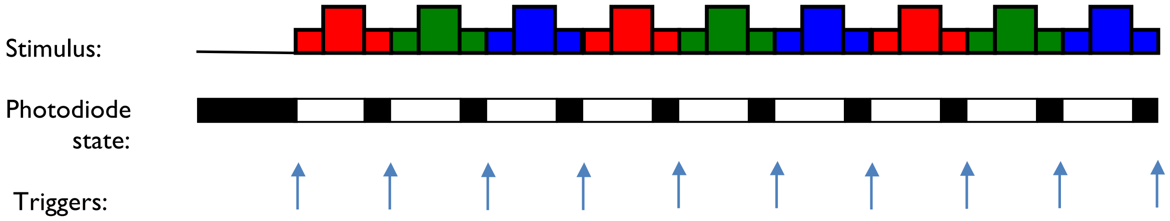
8. Baseline Time

As mentioned in section (1), when the script is run the screen will turn white, then grey when it is ready to start. Once a key has been pressed the screen will turn black. The screen remains black for a period of time defined by *baseLineTime*, in order to record spontaneous activity. After this time period the stimuli are shown in order.

When triggering is on, the script waits for the first trigger before showing the stimuli, rather than waiting a predefined time.

9. Photodiode support

VisStimAlex has built-in support for photodiode monitoring. This consists of a rectangle, the size and shape of which can be defined by the user. This rectangle displays as black when the program is waiting for a trigger, and white when it is running in its own time. So, for an HDH experiment with triggering, the photodiode indicator rectangle will be black during baseline, white for the first hold and the drift, and black during the second hold:



To enable the photodiode indicator rectangle, set ‘*photoDiode*’ to ‘on’, and specify the size of the rectangle in x and y using ‘*diodePatchXSize’* and ‘*diodePatchYSize*’. The patch is displayed in the bottom left corner of the screen.

No extra photodiode indication is provided for a *Flip* stimulus, as the photodiode can just record the stimulus itself. For a *D* stimulus, which has only one state per cycle, the photodiode indicator patch flashes white during the first frame, and then it returns to black for the rest of the frames in that direction

10. Parameter List

i. Programme behaviour parameters

keyWait	1 - the program waits for a keypress during the grey screens at the beginning and end. 0 is the default.
screenClear	1 – the program clears the screen following a keypress once the final grey screen is shown [default]. 0 – it doesn’t.

ii. Regular experimental parameters

experimentType	see section 4
triggering	see section 6
photoDiode	see section 9
gratingType	0 for a sine wave grating, 1 for a square wave grating
spaceFreqDeg	spatial frequency of stationary and drifting gratings, in cycles / deg
tempFreq	temporal frequency of drifting gratings, in Hz
directionsNum	How many different directions to display, starting at 0° and divided evenly clockwise
baseLineTime	see section 7
repeats	how many times to repeat the whole run of all directions
randMode	randomisation: 0: orderly, starting at 0° and proceeding clockwise 1: the same random permutation of directions is used in each repeat 2: a new random permutation of directions is used in each repeat 3: maximally different directions (180° plus a bit between each stimulus)
preDriftHoldTime	how long to display a stationary grating for before a drift (used in HD and HDH mode)
driftTime	How long to drift for (used in DH and HDH mode, and D and HD modes when triggering is off)
postDriftHoldTime	how long to display a stationary grating for after a drift (used in DH and HDH modes when triggering is off)
screenWidthCm	How wide the display screen is. Used to calculate how large the grating should be, based on desired spatial frequency
mouseDistanceCm	How far the mouse's eye is from the screen. Used to calculate how large the grating should be, based on desired spatial frequency
diodePatchXSize	see section 9
diodePatchYSize	see section 9

iii. System parameters

inputLine	The NI input line, for triggering
inputPort	The NI input port, for triggering
deviceName	The name of the NI board e.g. 'Dev1'